Cooperative Scene-Event Modelling for Acoustic Scene Classification

Yuanbo Hou, Bo Kang, Andrew Mitchell, Wenwu Wang, Jian Kang, Dick Botteldooren

***Abstract*—Acoustic scene classification (ASC) can be helpful for creating context awareness for intelligent robots. Humans naturally use the relations between acoustic scenes (AS) and audio events (AE) to understand and recognize their surrounding environments. However, in most previous works, ASC and audio event classification (AEC) are treated as independent tasks, with a focus primarily on audio features shared between scenes and events, but not their implicit relations. To address this limitation, we propose a cooperative scene-event modelling (cSEM) framework to automatically model the intricate scene-event relation by an adaptive coupling matrix to improve ASC. Compared with other scene-event modelling frameworks, the proposed cSEM offers the following advantages. First, it reduces the confusion between similar scenes by aligning the information of coarse-grained AS and fine-grained AE in the latent space, and reducing the redundant information between the AS and AE embeddings. Second, it exploits the relation information between AS and AE to improve ASC, which is shown to be beneficial, even if the information of AE is derived from unverified pseudo-labels. Third, it uses a regression-based loss function for cooperative modelling of scene-event relations, which is shown to be more effective than classification-based loss functions. Instantiated from four models based on either Transformer or convolutional neural networks, cSEM is evaluated on real-life and synthetic datasets. Experiments show that cSEM-based models work well in real-life scene-event analysis, offering competitive results on ASC as compared with other multi-feature or multi-model ensemble methods. The ASC accuracy achieved on the TUT2018, TAU2019, and JSSED datasets is 81.0%, 88.9% and 97.2%, respectively.**

***Index Terms*—Acoustic scene classification, Audio event classification, Scene-event relation, Cooperative modelling.**

I. Introduction

Acoustic scene classification (ASC) aims to tag an audio recording with predefined semantic labels that depict the environment in which the audio was recorded. Audio event classification (AEC) performs multi-label classification on an audio clip and aims to identify target events in the audio clip. ASC and AEC-related systems are used in various applications, such as medical surveillance [1] and video analysis [2].

In previous studies, such as [3], ASC and AEC are often treated as separate tasks, with models built independently for each task. However, acoustic scenes (AS) and audio events (AE) in natural environments usually accompany each other, and they are often implicitly associated. Certain AE may occur in a specific acoustic scene, while different AS may contain their representative events. For example, in the acoustic scene *park*, AE of *bird flight* and *dog barking* are likely to occur. Such fine-grained events form the basis of polyphonic AS. Humans use these fine-grained events and the overall acoustic background to understand and recognise their surrounding environment [4]. Motivated by the above observation, a few studies have proposed to analyse AS and AE jointly. For example, in [5], a simple but intuitive approach is introduced to perform ASC and AEC simultaneously by training a shared feature encoder and performing classification on latent embeddings. In addition, a synthesised dataset [5] is created to evaluate such studies by mixing foreground events with background scenes. Another line of studies adopts a different framework, which relies on shared low-level but separated high-level embedding spaces. Specifically, in [6], scene-event representations are learned with three shared convolutional layers based on multi-task learning (MTL), while high-level representations are learned without scene-event interaction. Using the MTL paradigm, a scene conditional-loss model [7] is used to learn the scene-to-event relation. Given that $P(AE|AS)$ denotes the probability that AE occurs in AS, $P(AE|AS)$ represents the scene-to-event relation that can be used to infer AE from AS, but this relation is one-way, which means that $P(AS|AE)$ cannot be inferred from $P(AE|AS)$. In relation-guided ASC (RGASC) [8], $P(AE|AS)$ is exploited on a fixed prior matrix. Unlike the MTL approach of sharing fixed layers, the cross-stitch method [9] connects fully connected and pooling layers of dual-task network branches, by learning improved representations of both tasks, through modelling shared representations with their linear combinations.

The above scene-event joint learning methods can be simplified to two frameworks: **m**odelling based on the same **o**ne **e**mbedding space (MoE) [5], and **m**odelling based on shared **l**ow-level and separated **h**igh-level **e**mbedding spaces (MlhE) [6][10][7]. MoE, also named hard parameter sharing [11], tries to obtain the same representations applicable to both AS and AE. However, AS and AE naturally follow a hierarchical relationship. As AEs are building blocks of AS, the coarse-grained scene and fine-grained event information have their own intrinsic properties. Therefore, MoE has limitations in capturing the intricate and changeable relation between AS and AE in real life. Unlike MoE, the MlhE framework, which

The WAVES Research Group received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme.

Yuanbo Hou and Dick Botteldooren are with the WAVES Research Group, Ghent University, Belgium (e-mail: {Yuanbo.Hou, Dick.Botteldooren}@UGent.be).

Bo Kang is with the IDLab, Ghent University, Belgium (e-mail: Bo.Kang@UGent.be).

Wenwu Wang is with the Centre for Vision, Speech, and Signal Processing, University of Surrey, Guildford, UK (e-mail: w.wang@surrey.ac.uk).

Andrew Mitchell and Jian Kang are with Institute for Environmental Design and Engineering, The Bartlett, University College London, UK (e-mail: {andrew.mitchell.18, j.kang}@ucl.ac.uk)

Corresponding author: Yuanbo Hou.

is similar to soft parameter sharing [11], exploits the shared scene-event and separated task-dependent representations for ASC and AEC, respectively. The joint learning in MlhE reduces the chances of overfitting, thereby resulting in improved performance for audio-related analysis tasks [12]. However, the MlhE framework does not fully utilize the inherent and implicit relation between AS and AE. Although the model [7] based on MlhE uses the fixed scene-to-event binary relation (i.e. presence or absence), such relation is derived from the synthetic dataset [5], and thus can be difficult to match with the complex scene-event relation in real life.

Real-world scene-event relations are not simply binary for presence or absence, but rather a likelihood expressed by probability. Humans can infer the ongoing AE from AS, and infer AS from AE. For example, there is a high probability of *birds singing* in the *park* scene, and the sound of *whizzing cars* often occurs in the *street* scene. That is, relations between AS and AE are typically two-way, instead of the fixed one-way scene-to-event relations in the conditional-loss model [7] and RGASC [8]. To exploit the two-way scene-event relation for ASC, this paper proposes a **c**ooperative **s**cene-**e**vent **m**odelling (cSEM) framework, where an adaptive coupling matrix is introduced for modelling the implicit scene-event two-way relations, i.e. $P(AE|AS)$ and $P(AS|AE)$. The coupling matrix thus acts as a two-way bridge for the mutual interaction and transformation between the high-level representations of AS and AE, reducing the overlap between semantic spaces of AS and AE, thus classifying AS and AE collaboratively. Different from RGASC [8], which uses a dataset-specific fixed prior matrix to crudely map the final predictions of ASC and AEC branches, the cSEM aims to automatically align the core knowledge extracted from AS and AE through a two-way scene-event relationship model in an end-to-end manner.

The main contributions of this work are summarized as follows: 1) We propose a novel framework cSEM for modelling the two-way scene-event relation and use it to improve the ASC performance. We instantiate the cSEM framework with Transformer-based and CNN-based models. 2) We conduct various experiments with detailed analysis, and further compare the cSEM-based models with the state-of-the-art models to illustrate the benefit of the cSEM framework. 3) To improve the understanding of the cSEM framework, we use visualization to provide insights into the capability of the cSEM framework in aligning the knowledge of AS and AE, and reducing redundant information between the AS and AE embeddings. Furthermore, we analyze the differences between real-life scenes from the perspective of events using the cSEM-based model for scene-event joint analysis.

This paper is organized as follows. Section II introduces scene-event joint modelling frameworks in prior studies and proposes the cSEM framework. Section III presents models based on the proposed cSEM framework. Section IV describes datasets and experimental setup. Section V analyses the results. Section VI draws conclusions.

II. Scene-Event Joint modelling Frameworks

In this section, we discuss two existing frameworks, namely MoE and MlhE, as introduced in Section I, and compare their similarities and differences. We then propose the cSEM framework, which has an advantage in exploiting the relation between scenes and events.

A. MoE: modelling using the same one embedding space

As shown in Fig. 1 (a), in MoE, the input time-frequency representations $X(t; f)$ are mapped by the encoder layers and transformed to the joint modelling space of AS and AE. The encoder layers can be formed as several types of neural networks, such as deep neural networks (DNN), convolutional neural networks (CNN), and recurrent neural networks (RNN) [13]. Let $\boldsymbol{X} \in \mathbb{R}^{T\times F}$ denote the time-frequency representations, where T and F denote the number of time frames and frequency bins. The encoder f turns $\boldsymbol{X}$ into an internal representation with dimension d_1, namely $f(\boldsymbol{X}) = \boldsymbol{R}_{se}$, where $\boldsymbol{R}_{se}$[1] $\in \mathbb{R}^{d_1}$ is the joint representation of AS and AE. Then, the scene classification layer maps $\boldsymbol{R}_{se}$ onto the target scene components through an adaptive weight matrix $\boldsymbol{W}_s$ and outputs the prediction of scenes $\hat{y}_s \in \mathbb{R}^{n_s}$,

$$\hat{y}_s = f_s(\boldsymbol{R}_{se}\boldsymbol{W}_s^T) = f_s(\hat{z}_s) \tag{1}$$

where $\boldsymbol{W}_s \in \mathbb{R}^{n_s\times d_1}$, n_s is the number of scene classes, f_s is the activation function, and $\hat{z}_s$ is the logit [14] of $\hat{y}_s$. Relying on the predicted $\hat{y}_s$ and the ground-truth label of scene y_s, the ASC loss can be defined as $\mathrm{L}_{\mathrm{scene}} = \mathrm{loss}_{\mathrm{s}}(\hat{y}_s, y_s)$. ASC is usually viewed as a single-label multi-class classification problem, hence the potential option of f_s is Softmax and $\mathrm{loss}_{\mathrm{s}}$ is cross entropy (CE) loss [15].

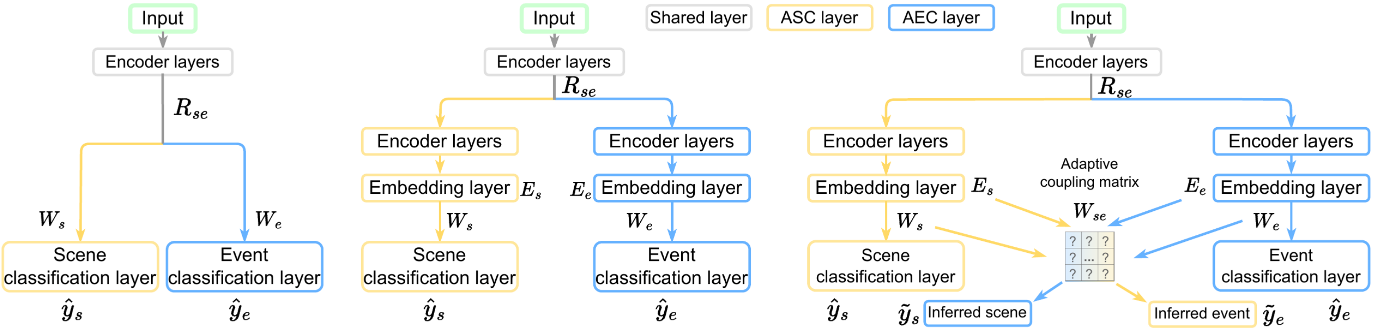
The event prediction $\hat{y}_e \in \mathbb{R}^{n_e}$ is obtained by projecting the internal representation onto the event classification layer through an adaptive weight matrix $\boldsymbol{W}_e \in \mathbb{R}^{n_e\times d_1}$, namely:

$$\hat{y}_e = f_e(\boldsymbol{R}_{se}\boldsymbol{W}_e^T) = f_e(\hat{z}_e) \tag{2}$$

where n_e is the number of event classes, f_e is the activation function, and $\hat{z}_e$ is the logit of $\hat{y}_e$. The AEC loss can be derived from the distance between the predicted $\hat{y}_e$ and the label of the event y_e, i.e. $\mathrm{L}_{\mathrm{event}} = \mathrm{loss}_{\mathrm{e}}(\hat{y}_e, y_e)$. AEC performs multi-label classification on audio clips, so the potential option for f_e is Sigmoid and the corresponding loss function $\mathrm{loss}_{\mathrm{e}}$ is binary cross entropy (BCE) [16]. Then, the final loss of MoE is $\mathrm{L} = \lambda_1\mathrm{L}_{\mathrm{scene}} + \lambda_2\mathrm{L}_{\mathrm{event}}$, where λ_i $(i = 1, 2)$ adjust the weights between the loss components, and λ_i default to 1.

An advantage of MoE is its simplicity and efficiency of learning joint representations $\boldsymbol{R}_{se}$ that are applicable to both AS and AE. This allows a robust and general feature extractor to be obtained, which can easily transfer the learned knowledge to related pattern recognition tasks [16]. However, in practice, AS and AE are represented in different levels of information in an audio clip. The AS and AE, which are from the clip level and frame level, respectively, are not only correlated, but also different in their characteristics. The models in [17][18][19] focus on the fine-grained AE information, and thus can not fully capture the coarse-grained AS information. Vice versa, the models in [20][21] focus on global AS features, and are limited in capturing the subtle differences between similar events. In short, MoE is limited in dealing with intricate real-world situations in the presence of varied acoustic scenes and diverse audio events.

[1] To simplify expressions, the batch size in the symbolic representation of the sample is omitted, i.e. $\mathbb{R}^{d_1}$ is $\mathbb{R}^{1\times d_1}$ in the training.



(a) MoE framework (b) MlhE framework (c) Proposed cSEM framework

Fig. 1. The three frameworks for scene-event joint modelling: (a) MoE: modelling using the same one embedding space; (b) MlhE: modelling using shared low-level and separated high-level embedding spaces; (c) cSEM: the proposed cooperative scene-event modelling with adaptive coupling matrix.

B. MlhE: modelling using shared low-level and separated high-level embedding spaces

As shown in Fig. 1 (b), in MlhE, separate encoding layers are further used to extract task-dependent acoustic representations of AS and AE. As found in [22], the low-level basic acoustic features are transferable to some extent and hence applicable in ASC and AEC tasks. However, high-level abstract representations are often difficult to be adapted to different scenarios. With the subsequent embedding layer, which usually consists of fully connected layers, the audio representations can be mapped to the high-level embeddings in the semantic space to be suitable for the corresponding classification tasks.

Following notations of MoE, the shared encoder turns the input $\boldsymbol{X}$ into joint scene-event representations $\boldsymbol{R}_{se}$. Next, the separated encoder further extracts the AS representations $\boldsymbol{R}_s$ and the AE representations $\boldsymbol{R}_e$, respectively. The subsequent embedding layers transform $\boldsymbol{R}_s$ and $\boldsymbol{R}_e$ into the embeddings of scenes $\boldsymbol{E}_s \in \mathbb{R}^{d_s}$ and the embeddings of events $\boldsymbol{E}_e \in \mathbb{R}^{d_e}$. Then, similar to the operations in MoE, the scene classification layer of MlhE maps $\boldsymbol{E}_s$ onto target scene components by $\boldsymbol{W}_s$ and outputs the prediction of scenes $\hat{y}_s \in \mathbb{R}^{n_s}$,

$$\hat{y}_s = f_s(\boldsymbol{E}_s \boldsymbol{W}_s^T) = f_s(\hat{z}_s) \tag{3}$$

where $\boldsymbol{W}_s \in \mathbb{R}^{n_s \times d_s}$, d_s is the dimension of scene embedding space, and $\hat{z}_s$ is the corresponding logit. Similarly, the prediction $\hat{y}_e$ of the event branch is obtained by

$$\hat{y}_e = f_e(\boldsymbol{E}_e \boldsymbol{W}_e^T) = f_e(\hat{z}_e) \tag{4}$$

where $\boldsymbol{W}_e \in \mathbb{R}^{n_e \times d_e}$, d_e is the dimension of the event embedding space, $\hat{y}_e \in \mathbb{R}^{n_e}$, and $\hat{z}_e$ is the logit. Finally, the losses of MlhE are the distance between the predicted value and the corresponding label, i.e. $\mathrm{L}_{\text{scene}} = \mathrm{loss_s}(\hat{y}_s, y_s)$ and $\mathrm{L}_{\text{event}} = \mathrm{loss_e}(\hat{y}_e, y_e)$. The final loss of MlhE is $\mathrm{L} = \lambda_1 \mathrm{L}_{\text{scene}} + \lambda_2 \mathrm{L}_{\text{event}}$, where λ_i $(i = 1, 2)$ is set typically to 1.

MlhE learns high-level task-goal-oriented representations based on low-level scene-event representations. Compared with MoE, MlhE has the advantage of utilizing richer information of shared and individual representations of coarse-grained AS and fine-grained AE. However, as MlhE does not explicitly coordinate the interaction between representations of AS and AE, the discriminative ambiguity remains for some similar audio clips. Intuitively, real-life audio clips may contain AS with overall similar sound, but they can be distinguished using implicit AE information specific to a scene class. For example, when the scene branch is uncertain about the scene label of an audio clip being *clamorous streets* or *noisy parks*, if the event branch indicates that the clip contains the audio event features of *birds singing* and *dogs barking*, then the scene branch can be more confident about the audio clip being from a *park* scene. Likewise, some audio clips may have similar sound events. In this case, the scene information implied in the contextual background can be used to clarify ambiguities caused by similar sound events. For example, when the event branch is uncertain about a clip whether it contains *cat meowing* or *baby crying*, but if the scene branch indicates that the audio clip is more likely to occur in a *nursery room*, then the clue from the scene branch can help the event branch to reduce its confidence on the prediction of the less likely event *cat meowing*. In short, MlhE neglects the implicit and intricate relation between scenes and events.

C. cSEM: proposed cooperative scene-event modelling

Different from MoE and MlhE, we present a novel cSEM framework, as shown in Fig. 1 (c).

In cSEM, a coupling matrix $\boldsymbol{W}_{se}$ is constructed to capture the bidirectional relation between AS and AE based on the core knowledge of AS and AE learned by the model, i.e. the weight matrices from the classification layers $\boldsymbol{W}_s$ and $\boldsymbol{W}_e$. This coupling matrix is used to map AS embeddings $\boldsymbol{E}_s$ to the event space to infer the corresponding event $\tilde{y}_e$. Then, the loss between the inferred event $\tilde{y}_e$ and the actual output of the AEC branch is calculated, which is back propagated to update the relevant weights used in the coupling matrix. Similarly, AE embeddings $\boldsymbol{E}_e$ are first mapped into the scene space to infer the corresponding scene output $\tilde{y}_s$, and then the loss between the inferred scene $\tilde{y}_s$ and the actual output of the ASC branch is measured to correct the learnable weights. In this process, $\boldsymbol{W}_{se}$ will model the implicit two-way scene-event relation, with which the ASC and AEC branches will collaborate to estimate each other's output and classify their targets.

Algorithm 1 shows the pseudo-code of cSEM. To model the bidirectional scene-event relations, inspired by self-attention [23], the dimensions of embedding spaces in Fig. 1 are set equal, i.e. $d_e = d_s$ and $\boldsymbol{W}_{se} \in \mathbb{R}^{n_s \times n_e}$. The role of coupling matrix $\boldsymbol{W}_{se}$ is to serve as a two-way scene-event bridge for the coordination and collaboration of the AS knowledge $\boldsymbol{W}_s$ and the AE knowledge $\boldsymbol{W}_e$ learned by the model, and to further reduce the extent of overlap between the ASC and the AEC branches in latent semantic space, resulting in a reduction of redundancy in the core information of AS and the core information of AE. The reduction of redundant information facilitates both branches to learn their target representations as much as possible while modelling their cooperative relations.

Algorithm 1 PyTorch pseudo-code of the cSEM framwork

```
1:  for $\boldsymbol{X}$ in Dataloader:                                  # $\boldsymbol{X}$: input acoustic feature
2:      $\boldsymbol{R}_{se}$ = $f(\boldsymbol{X})$                                  # $f$: shared_encoder_layers
3:      $\boldsymbol{E}_s$ = ASC_Embedding(ASC_encoder($\boldsymbol{R}_{se}$))
4:      $\boldsymbol{E}_e$ = AEC_Embedding(AEC_encoder($\boldsymbol{R}_{se}$))
5:      $\hat{y}_s = f_s(\boldsymbol{E}_s \boldsymbol{W}_s^T)$               # $\boldsymbol{W}_s$: scene_classification_layer.weight
6:      $\hat{y}_e = f_e(\boldsymbol{E}_e \boldsymbol{W}_e^T)$               # $\boldsymbol{W}_e$: event_classification_layer.weight
7:      $\mathrm{L}_{\text{scene}} = \mathrm{loss}_{\mathrm{s}}(\hat{y}_s, y_s)$, $\mathrm{L}_{\text{event}} = \mathrm{loss}_{\mathrm{e}}(\hat{y}_e, y_e)$

8:      $\boldsymbol{W}_{se}$=$\boldsymbol{W}_s \boldsymbol{W}_e^T$             # $\boldsymbol{W}_{se}$: adaptive scene-event coupling matrix
9:      $\boldsymbol{A}_e$=Softmax($\boldsymbol{W}_{se}$)                # $\boldsymbol{A}_e$: attention factor of audio events
10:     $\boldsymbol{K}_{e2s} = \boldsymbol{A}_e \boldsymbol{W}_e$           # $\boldsymbol{K}_{e2s}$: event-to-scene transformation matrix
11:     $\tilde{y}_s = \boldsymbol{E}_e \boldsymbol{K}_{e2s}^T$                         # $\tilde{y}_s$: inferred scene by event
12:     $\mathrm{L}_{\text{s\_by\_e}} = \mathrm{loss}_{\text{s\_by\_e}}(\tilde{y}_s, \hat{y}_s)$             # loss_function(input, target)

13:     $\boldsymbol{A}_s$=Softmax($\boldsymbol{W}_{se}^T$)              # $\boldsymbol{A}_s$: attention factor of acoustic scenes
14:     $\boldsymbol{K}_{s2e} = \boldsymbol{A}_s \boldsymbol{W}_s$           # $\boldsymbol{K}_{s2e}$: scene-to-event transformation matrix
15:     $\tilde{y}_e = \boldsymbol{E}_s \boldsymbol{K}_{s2e}^T$                         # $\tilde{y}_s$: inferred event by scene
16:     $\mathrm{L}_{\text{e\_by\_s}} = \mathrm{loss}_{\text{e\_by\_s}}(\tilde{y}_e, \hat{y}_e)$             # loss_function(input, target)

17:     $\mathrm{L} = \lambda_1 \mathrm{L}_{\text{scene}} + \lambda_2 \mathrm{L}_{\text{event}} + \lambda_3 \mathrm{L}_{\text{s\_by\_e}} + \lambda_4 \mathrm{L}_{\text{e\_by\_s}}$    # final loss
```

Lines 9 to 12 in Algorithm 1. To explore the possibility of inferring AS using AE embeddings, the weights in $\boldsymbol{W}_{se}$, which indicate the distribution of all AEs in each scene, are first calculated by *row-wise* Softmax [23] to obtain the attention factor of events $\boldsymbol{A}_e \in \mathbb{R}^{n_s \times n_e}$ that assigns a weight to each event in the corresponding scene. Using $\boldsymbol{A}_e$, the learned event knowledge $\boldsymbol{W}_e$ is transformed into the scene space to obtain the event-to-scene knowledge transformation matrix $\boldsymbol{K}_{e2s} \in \mathbb{R}^{n_s \times d_e}$. Finally, based on $\boldsymbol{K}_{e2s}$, the corresponding inferred scene by event, $\tilde{y}_s \in \mathbb{R}^{n_s}$, can be derived from AE embeddings $\boldsymbol{E}_e$ via the adaptive $\boldsymbol{W}_{se}$. The loss $\mathrm{L}_{\text{s\_by\_e}}$ between the inferred scene $\tilde{y}_s$ and the actual output $\hat{y}_s$ of ASC branch can be fed back to the AEC branch to update relevant weights, which further improves the quality of $\boldsymbol{W}_{se}$ and better captures the implicit and intricate scene-event relation.

Lines 13 to 16 in Algorithm 1. The process of inferring AE from AS embeddings $\boldsymbol{E}_s$ is similar to inferring scenes from events described above. First, the attention factor of scenes $\boldsymbol{A}_s \in \mathbb{R}^{n_e \times n_s}$ is computed where the weights are assigned to each scene for a given event. Then, the learned scene knowledge $\boldsymbol{W}_s$ is transformed into the event space by multiplying $\boldsymbol{A}_s$ resulting in the scene-to-event knowledge transformation matrix $\boldsymbol{K}_{s2e} \in \mathbb{R}^{n_e \times d_s}$. Finally, using $\boldsymbol{K}_{s2e}$, the corresponding inferred event by scene, $\tilde{y}_e \in \mathbb{R}^{n_e}$, can be derived from $\boldsymbol{E}_s$. The loss $\mathrm{L}_{\text{e\_by\_s}}$ between the inferred event $\tilde{y}_e$ and the actual output $\hat{y}_e$ of the AEC branch is fed back to the ASC branch to update the relevant weights. The coupling matrix-related code, shown from 8 to 16 lines in Algorithm 1, is the only extra part of cSEM, compared to MlhE.

In the cSEM framework, the loss functions $\mathrm{loss}_{\text{s\_by\_e}}$ and $\mathrm{loss}_{\text{e\_by\_s}}$ model the scene-event relation and similarity between the derived results and the actual outputs. That is, they aim to match another branch's output. Hence, mean squared error (MSE), which performs well in regression tasks [24], is chosen for these loss functions. If the inferred scene by event, $\tilde{y}_s$, is regarded as the predicted scene vector in the semantic space. The output of the ASC branch, $\hat{y}_s$, is viewed as the actual scene vector in the semantic space. The goal of MSE is to measure the absolute distance between the two vectors in the latent space, while the classification loss, like CE, is used to measure the difference between classification results of each class in the two vectors. Meanwhile, to improve the classification accuracy, CE tends to expand the distance between the target output and other non-target outputs, that is, to enlarge the distance between different classes [25]. Thus, the regression loss such as MSE is apt to consider the whole output derived from embeddings as the target to optimize, while the classification loss such as CE optimizes the class-wise loss to improve the classification accuracy and enlarge the gap between the target class and the non-target class. Comparison of the model performance achieved using several options for $\mathrm{loss}_{\text{s\_by\_e}}$ and $\mathrm{loss}_{\text{e\_by\_s}}$ can be found in Section V-D.

Combining the loss of the ASC, the AEC, the inferred scene by event, and the inferred event by scene, the loss of the cSEM read as $\mathrm{L} = \lambda_1 \mathrm{L}_{\text{scene}} + \lambda_2 \mathrm{L}_{\text{event}} + \lambda_3 \mathrm{L}_{\text{s\_by\_e}} + \lambda_4 \mathrm{L}_{\text{e\_by\_s}}$, where λ_i $(i = 1, 2, 3, 4)$ is the scale factor of each loss, set empirically to 1. Compared with MlhE, cSEM has an advantage in that it learns the coupling matrix $\boldsymbol{W}_{se}$, which has the potential to capture the implicit relation between the real-life varied scenes and diverse events. With the two-way collaborative scene-event interaction, the cSEM framework can further facilitate the downstream classification tasks.

III. Instantiations of the Proposed cSEM

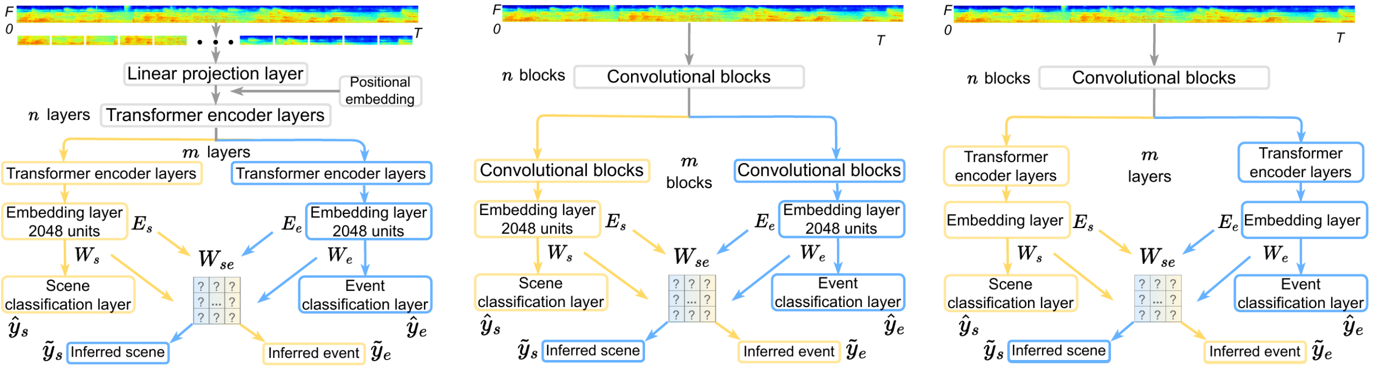
In ASC and AEC tasks, the most commonly used network before is CNN [26], but now Transformer [23] is gradually taking over. This section will show four instantiations of cSEM based on Transformer and CNN models.

A. cSEM-AST: cSEM-based Audio Spectrogram Transformer

Audio Spectrogram Transformer (AST) [27] has recently achieved competitive results on audio classification tasks on AudioSet [28]. To alleviate the tendency of overfitting in Transformer models [29], the AST with 10 encoder layers is used in this paper, instead of the default 12 encoder layers. Fig. 2 (a) illustrates the proposed cSEM-based AST (cSEM-AST).

1) Shared parts: First, the audio waveform is converted into a spectrogram. The spectrogram is then split into a sequence of patches, and each patch is flattened to an embedding using a linear projection layer. The patch sequence does not keep the temporal order, and Transformer does not capture the input order information. Thus, a trainable positional embedding is added to each patch embedding to allow the model to preserve the temporal order of patches. The total number of encoder layers used here in AST is 10. Assuming that n shared encoder layers are in Fig. 2 (a), the remaining $m = 10 - n$ layers will learn the individual task-dependent representations separately.

2) Separated parts: To extract the individual high-level task-dependent representations for ASC and AEC, the scene-event representations are fed into the encoder layers of the ASC and AEC branches, respectively. The learned representations are then fed into the embedding layer to learn


F
0
T
Linear projection layer
Positional embedding
n layers
Transformer encoder layers
m layers
Transformer encoder layers
Transformer encoder layers
Embedding layer 2048 units
Embedding layer 2048 units
E_s
E_e
W_s
W_e
W_{se}
Scene classification layer
Event classification layer
$\hat{y}_s$
$\hat{y}_e$
$\tilde{y}_s$ Inferred scene
Inferred event $\tilde{y}_e$
n blocks
Convolutional blocks
m blocks
Convolutional blocks
Convolutional blocks
m layers
Transformer encoder layers
Embedding layer

(a) cSEM-AST (b) cSEM-PANN (c) cSEM-CNNT

Fig. 2. Models based on the proposed cSEM framework.

improved mappings in the latent semantic space that will be later used for classification. The weights $\boldsymbol{W}_s$ and $\boldsymbol{W}_e$, which contain highly condensed information about similarities and differences between different targets, will be used to infer each other's outputs under the guidance of the scene-event relation module, and to refine their respective estimations.

3) Modelling of scene-event relation: The relation module aims to align AS embeddings $\boldsymbol{E}_s$ with AE embeddings $\boldsymbol{E}_e$, and then map the core knowledge $\boldsymbol{W}_s$ and $\boldsymbol{W}_e$ about targets learned by the classification layers from different semantic spaces into the scene-event joint space $\boldsymbol{W}_{se}$, which is dedicated to modelling the implicit relation between AS and AE without any prior knowledge. The process for learning the coupling matrix in cSEM is driven by the losses between the derived results $\tilde{y}_e$ and $\tilde{y}_s$ and the actual predictions $\hat{y}_e$ and $\hat{y}_s$. The joint modelling of the scene-event relation enables explicit interaction between high-level representations from ASC and AEC branches. With such joint modelling, the latent semantic spaces of AS and AE will be gradually aligned.

B. cSEM-PANN: cSEM-based Pretrained Audio Neural Networks

To show the flexibility of the cSEM framework, we also propose cSEM based on the Pretrained Audio Neural Networks (PANN) [16], as shown in Fig. 2 (b). The differences between cSEM-PANN and cSEM-AST are that: (1) the spectrogram will be fed directly into cSEM-PANN, instead of slicing the spectrogram into patches as in cSEM-AST. The inputs of PANN are clip-level features, not patch-level. Thus, cSEM-PANN does not require the projection layer, the position embedding layer and the additional token as in AST; (2) Both the shared and branched layers in cSEM-PANN consist of typical convolutional blocks, which are based on stacked convolutional layers; (3) The total number of convolutional blocks in PANN is 6 [16], assuming that there are n shared blocks in Fig. 2 (b), the remaining $m = 6-n$ blocks will learn separate representations. Except for the input, the composition, and the number of layers, the remaining components of cSEM-PANN are the same as those of cSEM-AST.

C. cSEM-CNNT: cSEM-based CNN-Transformer

Transformer-based AST and convolution-based PANN have achieved excellent audio classification performance, so this paper tries to combine convolution and Transformer to exploit CNN's local feature extraction capabilities and Transformer's long-term context capture capabilities. To this end, a simple CNN-Transformer (CNNT) is proposed and instantiated based on the cSEM, as shown in Fig. 2 (c). Specifically, CNNT contains 3 convolutional blocks from PANN, a Transformer encoder layer from AST, an embedding layer and a classification layer. The features captured from the Transformer encoder layer are passed to the embedding layer. The default output size of the Transformer encoder layer is 512, so the number of units in the embedding layer in CNNT is also 512 by default.

D. cSEM-TinyCNN: cSEM-based TinyCNN

To further evaluate the cSEM, we propose TinyCNN with only 2 convolutional layers and 2 multi-layer perceptrons (MLP), and instantiate it based on cSEM. The convolutional layers in TinyCNN contain 32 and 64 filters with a kernel (7 × 7), respectively. The first MLP acts as an embedding layer with only 100 units, and the second MLP acts as a classification layer. The lightweight cSEM-TinyCNN explores the benefits of cSEM on ASC tasks for small models. The cSEM-TinyCNN has a similar but simpler structure to the cSEM-CNNT. Due to space constraints, cSEM-TinyCNN is not shown in Fig. 2.

IV. Datasets and Experimental Setup

This paper uses two real-life datasets and one synthetic dataset. (1) **Real-life datasets**. The TUT Urban Acoustic Scenes 2018 (TUT2018) dataset [15] contains 8640 10-second segments, in total, 24 hours of audio of 10-class scenes. The TAU Urban Acoustic Scenes 2019 (TAU2019) dataset [30] contains 14400 10-second audio segments, totalling 40 hours of audio of 10-class scenes. There are no event labels in scene datasets TUT2018 and TAU2019. Thus, pretrained models (AST and PANN) are used to tag each audio clip with pseudo labels to indicate the probabilities of the corresponding AE.
(2) **Synthetic dataset**. The joint sound scene and event dataset (JSSED) [5] has 3000 30-second segments. The JSSED consists of synthesized audio clips with 32-class AE and 10-class AS, where the events present in an audio clip are related to the scene of the clip. For each of the 10-class AS, there are 10 unique locations for each class, with a total of 100 background recordings. The AE to background AS signal-to-noise ratio is randomly assigned in the range -15 to 15 dB [5].

Pretraining. Most systems used as references in later experiments are from DCASE2018/2019 challenges [15] which allow the use of AudioSet and other audio scene datasets. AST and PANN perform excellently after they are trained on AudioSet [28]. Previous work shows that a large and deep model like AST performs poorly when trained on small datasets alone [31]. Therefore, this paper uses pretrained weights of PANN. As the AST used here has only 10 layers, the first 10-layer weights of the pretrained AST are used. For the proposed CNNT, its convolution part refers to the corresponding part of PANN. Hence, the first 3 convolutional blocks of CNNT are initialized with the weights of the corresponding convolutional blocks from the pretrained PANN, while the remaining encoder layer, embedding layer, and classification layer are randomly initialized without pretraining. The TinyCNN is pretrained for 10 epochs in the balanced subset of 22K audio clips of AudioSet. A batch size of 64, and an Adam [32] optimizer with a learning rate of 0.001 are used. To further explore the performance of the four models with (w/) and without (w/o) cSEM, Section V-B presents comprehensive results for each model w/ and w/o pretraining and w/ and w/o the proposed cSEM framework.

Experimental setup. The log Mel filterbank (fbank) is used as the acoustic feature [16]. The inputs to cSEM-AST differ slightly from those to cSEM-(PANN, CNNT, TinyCNN). For cSEM-AST, the audio clip is converted into a sequence of 128-dimensional fbank computed with the $25ms$ Hamming window and a hop size of $10ms$, then the spectrogram is split into a sequence of patches following the settings of [27]. For cSEM-(PANN, CNNT, TinyCNN), the number of mel filter banks is 64 [16]. Then the 64-dimensional fbank is extracted by STFT with Hamming window length of $46ms$ and overlap of $1/3$ between windows following the settings of [33]. The cSEM-based models are trained for a maximum of 100 epochs. Gradient accumulation with a batch size of 64, and an Adam optimizer [32] with initial learning rates of 1e-6 [27] and 1e-3 [33] are used to minimize losses in cSEM-AST and cSEM-(PANN, CNNT, TinyCNN), respectively. To prevent over-fitting, dropout [34], and normalization are used. Systems are trained on GPU cards Tesla V100 without a fixed seed. To facilitate the comparison of results with other systems, the training/testing split of the TUT2018 and TAU2019 datasets follow the default split of the DCASE2018 Task1A[2] and DCASE2019 Task1A[3]. In training, 20% of the training samples are randomly selected to form the validation set. For JSSED, as in [7], 2400 30-second audio clips are used for training, 300 for validation and 300 for testing. There is no overlap between the training, validation, and testing sets. Each system is run 10 times. The accuracy (Acc.) [33] is used as the metric. A larger Acc. indicates a better performance.

V. Results and Analysis

Although the synthetic JSSED has ground-truth labels of AS and AE, the diversity of AE, the complexity of AS, and the intrinsic logical relationships between AE and AS are inferior to those of the real-life TUT2018 and TAU2019. Hence, most of the experiments will be performed on real-life datasets. This section analyzes the performance of cSEM by following the research questions (RQs). RQ1-4 explore the performance of the proposed cSEM-based models from different perspectives, and compare the differences between the cSEM framework and other scene-event joint modelling frameworks. RQ5-6 provide intuitive insights into cSEM's capability in aligning the core knowledge of AS and AE, and applying the cSEM-based model to real-life scene-event analysis. RQ7 compares the cSEM with other scene-event joint analysis methods.

A. RQ1: Does more information shared in the proposed cSEM framework lead to better model performance?

The first step is to explore the ratio between the number of shared and separated layers of branches in the cSEM framework to determine the model structure for experiments. That is, how much of the separated task-oriented individual representations of AS and AE in different latent semantic spaces should be retained? The impact of the number of shared layers/blocks on cSEM-AST/PANN is explored in Table I.

TABLE I
ASC accuracy (%) of models with different numbers (n) of shared layers/blocks on the TUT2018 validation set.

n	cSEM-PANN	cSEM-AST	n	cSEM-PANN	cSEM-AST
1	78.03 ± 1.85	81.46 ± 0.92	6	75.64 ± 1.89	82.12 ± 1.77
2	**78.67** ± 1.31	81.91 ± 1.44	7	None	81.86 ± 1.18
3	78.13 ± 1.78	82.33 ± 1.39	8	None	81.59 ± 1.93
4	77.64 ± 1.95	**82.49** ± 1.52	9	None	81.43 ± 1.48
5	76.35 ± 0.93	82.30 ± 1.45	10	None	81.03 ± 1.28

As shown in Table I, the number of shared layers/blocks does not have a monotonous effect on the results of either cSEM-AST or cSEM-PANN. At first, the classification accuracy of models increases with the number of shared layers/blocks, but after reaching the peak, it starts to decrease with the increase in the number of shared layers/blocks. cSEM-AST achieves the best result when the first 4 layers are shared, while cSEM-PANN obtains the best result when the first 2 blocks are shared. In Table I, especially for cSEM-AST, the difference in the number of shared layers leads to a slight difference in the results. The reason may be that the multi-head attention and feed-forward layers in Transformer encoder [23] in the AST both contain residual connections. During the forward propagation of the network, the residual structure can enable the input signal to be propagated directly from any lower layer to the upper layer to prevent the problem of network degradation [35]. With residual connections, cSEM-AST becomes insensitive to the effect of the changing number of shared layers. In contrast, PANN does not have the residual structure. As a result, changes in the number of shared blocks have a larger impact on cSEM-PANN. The first 4 and 2 shared layers/blocks are used as the default configuration structures for cSEM-AST and cSEM-PANN, respectively. The results from cSEM-CNNT and cSEM-TinyCNN have similar trends to those from cSEM-PANN, and are not listed in detail due to space constraints. The best-shared layers for cSEM-CNNT and cSEM-TinyCNN are the first 2 and 1 layers. Subsequent experiments will be conducted on these structures.

[2] http://dcase.community/challenge2018/task-acoustic-scene-classification
[3] https://dcase.community/challenge2019/task-acoustic-scene-classification

B. RQ2: Does pretraining improve the performance of models? How do the scene-event joint frameworks MoE, MlhE, and the proposed cSEM perform on the same base model?

As shown in Table II, without pretraining, CNN-based PANN, CNNT, and TinyCNN outperform the Transformer-based AST for ASC on the real-life dataset. This is consistent with the results reported in a previous paper [31], which shows that Transformer-based models perform poorly on small datasets. Transformer-based models usually have more layers, which tend to overfit severely on small datasets [29] due to the large number of parameters involved in the model. After the AST is pretrained with the large-scale AudioSet, its performance is significantly improved. Among the 3 CNN-based models in Table II, TinyCNN has the smallest number of layers and the simplest structure, resulting in the worst scene classification performance. The convolutional blocks of CNNT are the same as those of PANN. However, CNNT with one Transformer encoder is slightly better than PANN results w/ or w/o pretraining, which implies that adding a Transformer encoder with attention to a pure convolutional model is beneficial in capturing the global context for ASC.

TABLE II
ACC. (%) FOR ABLATION STUDY OF THE EFFECT OF cSEM FRAMEWORK AND PRETRAIN ON ASC ON TUT2018 TEST SET.

#	Pretrain	cSEM	AST	PANN	CNNT	TinyCNN
1	✗	✗	60.29±0.55	69.85±0.71	70.88±0.92	67.20±0.43
2	✗	✓	61.72±0.82	72.21±0.60	73.34±0.79	68.92±0.67
3	✓	✗	76.98±0.95	74.35±0.74	75.74±1.09	71.32±1.43
4	✓	✓	**79.12**±0.89	**76.41**±0.83	**76.76**±0.94	**73.41**±1.17

The results of w/ and w/o cSEM framework in Table II show that cSEM, which aims to fuse fine-grained event with coarse-grained scene information, can help improve the accuracy of the corresponding model in ASC. The ASC results in Table II demonstrate the effectiveness of the cSEM framework in improving ASC performance via the analysis of scenes from the perspective of scene-event cooperative modelling.

TABLE III
ACC. (%) OF ASC AND AUC OF AEC RESULTS OF THREE FRAMEWORKS BASED ON THE SAME BACKBONE MODEL AST ON TEST SET.

Dataset	Task	MoE	MlhE	Proposed cSEM
TUT2018	ASC (%)	71.43 ± 1.44	77.57± 0.91	**79.12** ± 0.89
	AEC	0.977 ± 0.012	0.985 ± 0.007	0.986 ± 0.005
JSSED	ASC (%)	93.11± 0.87	94.14± 0.93	**94.97** ± 0.96
	AEC	0.989 ± 0.008	0.991 ± 0.008	0.990 ± 0.007

Inspired by the good performance of AST in Table II, AST is used as the base model to evaluate the performance of scene-event joint analysis frameworks. Table III shows the ASC and AEC results of the frameworks. The performance of MoE is inferior to the performance of MlhE, where low-level basic joint representations and high-level task-dependent individual representations are learned to improve adaptability and reduce potential overfitting of the model [12]. From another perspective, the underlying assumption of MoE is that the latent target semantic spaces of AS and AE are entirely consistent. The underlying assumption of MlhE is that, given the natural connections between AS and AE, the latent semantic spaces of AS and AE partially overlap, while the specific characteristics of these two targets should be different. Compared to MoE, the underlying assumption of MlhE tends to resemble the actual real-world situation more closely, thereby leading to better results in Table III.

The assumption of cSEM is that joint representations can be shared, and individual task-oriented representations can be associated to jointly model the implicit relation between AS and AE by the shared coupling matrix. The ASC and AEC branches in cSEM learn the similarities and differences between them by modelling the two-way scene-event relation. Hence, cSEM has a better capability of distinguishing similar scenes. Even though some results in Table III are close, the statistics of results of 10 runs of MlhE and cSEM in Table III on TUT2018 reveal that the cSEM provides a statistically significant improvement in ASC accuracy compared to MlhE.

The AEC results are also listed in Table III. To measure the performance of models for discriminating between events, the threshold-free AUC [36] is used. It can be observed that almost all models achieved excellent performance on the AEC of the used datasets, and the difference in the AEC results by various frameworks is relatively small. This is also consistent with the results in DCASE challenges [3], where most deep learning-based models achieved good AEC results.

TABLE IV
ACC. (%) OF ASC RESULTS OF DIFFERENT FRAMEWORKS BASED ON THE SAME BACKBONE MODEL AST ON JSSED TEST SET.

#	AEC labels used in training	MoE	MlhE	cSEM
1	GT labels with 80% Acc.	90.83±1.71	91.17±1.55	92.47±1.46
2	Pseudo labels with 80% Acc.	88.47±1.58	90.62±1.44	91.35±1.84

Table III uses pseudo labels of AE on TUT2018 and ground-truth (GT) labels of AE on JSSED. To examine whether pseudo labels similar to target augmentation offer additional benefits compared to real labels, Table IV uses GT labels from JSSED and pseudo labels from a pretrained model (both labels with 80% AEC accuracy on JSSED) to replace the AE labels of the training set in JSSED in Table III. AST-based frameworks are retrained to evaluate the impact of using pseudo and GT labels of AE on the ASC task. Table IV shows that the models using GT labels outperform models with pseudo labels, no matter what the framework is. That means, GT labels are more powerful for ASC than pseudo labels from the pretrained models.

TABLE V
PARAM. AND ENERGY-COSTLY MACS OF PRIMARY MODELS.

Task	Only ASC		Joint ASC and AEC			
Model	AST	PANN	MoE-AST	MlhE-AST	cSEM-AST	cSEM-PANN
Param.(M)	73.61	79.69	74.69	118.79	118.79	160.19
MACs.(G)	108.999	6.439	109.001	174.258	174.292	11.009

Table V presents the number of parameters (in Millions) and the multiply-accumulate operations (MACs in Gigas). Compared with MlhE, cSEM, which models two-way scene-event relations by the coupling matrix, does not introduce new parameters. Thus, the number of parameters of MlhE-AST and cSEM-AST are equal. Under the same task and backbone model, cSEM-AST only increases MACs by 0.034 G over MlhE-AST. Table VI presents MlhE-plus, which adds another embedding layer on the ASC and AEC branches of M1hE, to explore whether deepening the M1hE can improve its performance. The MlhE-plus-PANN results show that the newly

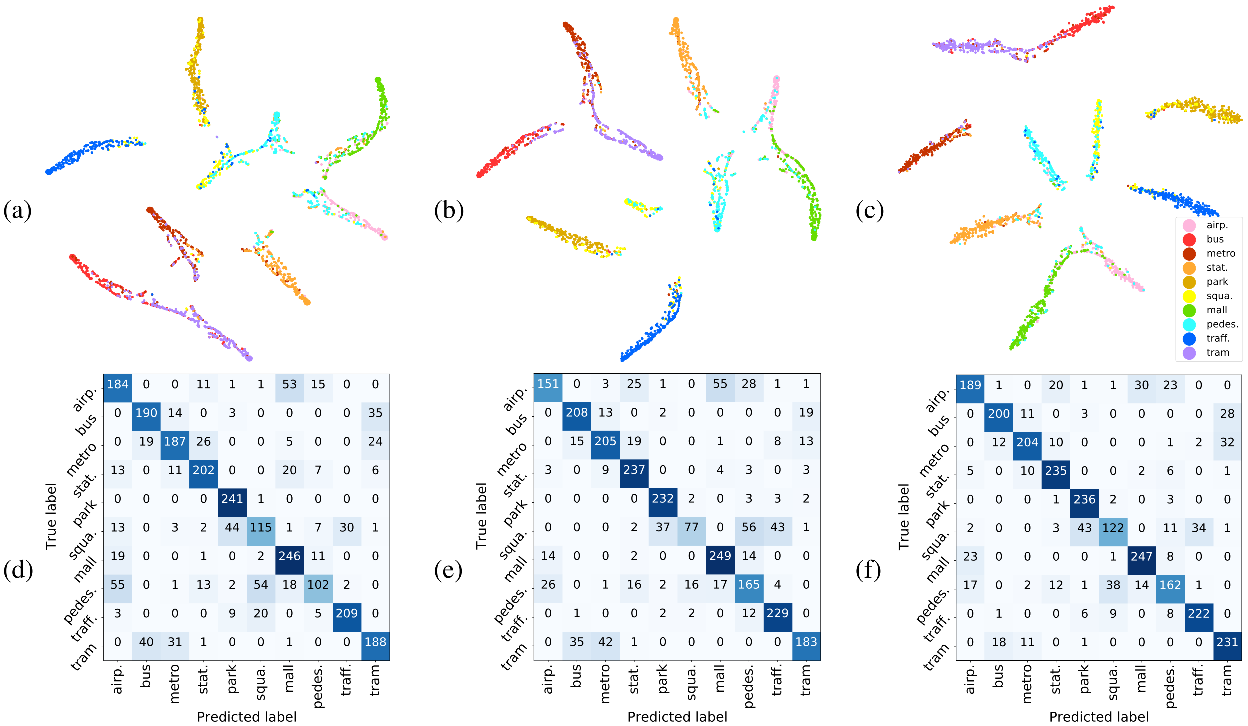


Fig. 3. Visualization of the learned representations using t-SNE [37]. Subplots (a), (b), and (c) are the output of the scene classification layer in *MoE-AST*, *MlhE-AST*, and *cSEM-AST* on TUT2018 test set, respectively. Subplots (d), (e), and (f) are the corresponding confusion matrix of (a), (b), and (c), respectively.

added parameters in MlhE-plus increase the computational overhead of the model, but do not improve its performance.

TABLE VI
ASC Acc. of different frameworks on TUT2018 test set.

Model w/ pretraining	MlhE-PANN	MlhE-plus-PANN	cSEM-PANN
Param. (M)	160.19	168.58	160.19
MACs. (G)	10.976	10.984	11.009
Acc. (%)	75.00±1.35	74.68±1.42	76.41±0.83

A visualization of the learned AS representations is shown in Fig. 3. In Fig. 3 (a), it can be seen that samples from street pedestrian (*pedes.*) are easily confused with those from shopping mall (*mall*) and public square (*squa.*). Samples from *tram* are mixed with samples from *bus* and *metro*. In Fig. 3 (b), samples of *tram* are easily compounded with samples of *metro*. Even for the human auditory system, it is challenging to distinguish these similar scenes by relying on audio only. The 10 classes of scenes are more clearly visible in Fig. 3 (c). The scenes, which can be easily confused in Fig. 3 (a) and (b), are distinguishable in Fig. 3 (c). In particular, *bus* can be clearly distinguished from *tram* and *metro*. The obscure samples of airport (*airp.*) shown in (a) and (b) are clustered into a distinct subclass in (c). The confusion matrices illustrate a similar but more pronounced boosting effect by the proposed cSEM.

C. RQ3: What effect do different values of λ have on the performance of cSEM-based models?

The loss weight λ represents the importance of its target in the overall model's performance. The proposed cSEM contains two types of losses: classification losses (L_{scene}, L_{event}) and regression losses ($L_{\text{s\_by\_e}}$, $L_{\text{e\_by\_s}}$). If the weights of classification losses are increased, the model will pay more attention to exploring ASC and AEC separately. If the weights of regression losses are increased, the model will be driven towards improved modelling of the scene-event relation in the joint space and the alignment of the AS embedding space with the AE embedding space.

Table VII first investigates the impact of losses on ASC (#1-#9), then the optimal ratio of fusing different losses (#10-#14). The λ_1, λ_2, λ_3, and λ_4 correspond to L_{scene}, L_{event}, $L_{\text{s\_by\_e}}$, and $L_{\text{e\_by\_s}}$, respectively. The AS information alone is exploited in #1, which can be regarded as a pure ASC model based on AST without the aid of additional information. #2 is effectively trained only for the AEC branch in cSEM-AST, so the output of its untrained ASC branch is random, resulting in poor performance. #3, #4 and #5 use $L_{\text{s\_by\_e}}$, $L_{\text{e\_by\_s}}$, and both of them as supervised loss for the dual-branch model in training. The labels in $L_{\text{s\_by\_e}}$ and $L_{\text{e\_by\_s}}$ come from the output of ASC and AEC branches, respectively, that is, their labels are pseudo labels from the model's own outputs instead of real labels. Therefore, the cases #3, #4 and #5 correspond to self-supervised learning. In #6, there are ASC and AEC branches, but no learning about the scene-event relation. #7 and #8 introduce $L_{\text{s\_by\_e}}$ and $L_{\text{e\_by\_s}}$ respectively, where $L_{\text{e\_by\_s}}$ facilitates the training of ASC branch, while $L_{\text{s\_by\_e}}$ is beneficial to AEC branch. The loss combination of #7 focuses on capturing global AS information, while #8 focuses more on AE representations. The performance of #7 is slightly better than that of #8, indicating that the information related to AS is more beneficial to ASC. #9 is a default coefficient combination, and is also used in RQ1 and RQ2. The results (#1-#9) indicate that in this task, the information represented by $\mathrm{L_{scene}}$ is crucial for cSEM-AST. The learning based on $\mathrm{L_{s\_by\_e}}$, $\mathrm{L_{event}}$, and $\mathrm{L_{e\_by\_s}}$ can complement the learning based on $\mathrm{L_{scene}}$ to further improve the accuracy of scene classification. Next, #10-#14 focus on exploring how to more effectively fuse these losses and maximize the benefits of using the information from events with noisy pseudo labels. It can be seen that giving a maximum weight to $\mathrm{L_{scene}}$ and a second-large weight to $\mathrm{L_{event}}$, while incorporating the scene-event relation information ($\mathrm{L_{s\_by\_e}}$ and $\mathrm{L_{e\_by\_s}}$) with smaller weights, leads to the best result as shown in #13. The ASC accuracy of #13 on the TUT2018 test set is **80.95%**±0.97%. Compared to AST in #3 of Table II, the best ASC accuracy of cSEM-AST on the test set is increased by 3.97%.

TABLE VII
THE EFFECT OF DIFFERENT VALUES OF λ IN CSEM-AST ON ACOUSTIC SCENE CLASSIFICATION ON THE VALIDATION SET OF TUT2018.

#	λ_1	λ_2	λ_3	λ_4	*Acc. (%)*	#	λ_1	λ_2	λ_3	λ_4	*Acc. (%)*
1	1	0	0	0	78.83±1.33	8	1	1	1	0	80.41±1.64
2	0	1	0	0	15.97±1.70	9	1	1	1	1	82.49±1.52
3	0	0	1	0	11.77±1.84	10	1	1	0.5	1	81.95±1.74
4	0	0	0	1	11.82±2.03	11	1	1	1	0.5	81.57±1.58
5	0	0	1	1	13.01±2.69	12	1	1	0.01	0.01	83.62±2.05
6	1	1	0	0	80.06±1.89	13	1	0.5	0.01	0.01	**83.81**±1.89
7	1	1	0	1	80.73±1.47	14	1	0.1	0.1	0.1	83.11±1.82

The results of cSEM-PANN in Table VIII show a similar trend to Table VII. #1-#9 in Table VIII show that the more the model pays attention to AS-related information, the better its performance. The best coefficient combination for cSEM-PANN is #14, and the corresponding accuracy on the TUT2018 test set is **78.50**%±0.76%, giving a 4.15% increase, as compared to PANN in #3 of Table II. Overall, cSEM-AST outperforms cSEM-PANN in the same framework, which implies that Transformer-based AST is more powerful than CNN-based PANN for audio classification-related tasks when large-scale data are used for training the models.

TABLE VIII
THE EFFECT OF DIFFERENT VALUES OF λ IN CSEM-PANN ON ACOUSTIC SCENE CLASSIFICATION ON THE VALIDATION SET OF TUT2018.

#	λ_1	λ_2	λ_3	λ_4	*Acc. (%)*	#	λ_1	λ_2	λ_3	λ_4	*Acc. (%)*
1	1	0	0	0	76.81±1.09	8	1	1	1	0	77.71±1.14
2	0	1	0	0	18.19±1.42	9	1	1	1	1	78.67±1.31
3	0	0	1	0	11.68±2.19	10	1	1	0.5	1	78.23±1.34
4	0	0	0	1	12.05±3.25	11	1	1	1	0.5	77.46±1.61
5	0	0	1	1	13.46±3.17	12	1	1	0.1	0.1	79.23±1.95
6	1	1	0	0	77.12±1.36	13	1	1	0.01	0.01	79.36±1.78
7	1	1	0	1	77.90±0.92	14	1	0.5	0.01	0.01	**79.80**±1.64

With the results in Table VII and Table VIII, we can draw the following observations: 1) The cSEM-based joint classification model outperforms the pure scene classification model for the ASC task. That is, AE information is helpful for improving AS classification. 2) Modelling the two-way scene-event relation and aligning the AS and AE embedding spaces benefit the scene-event classification. 3) The cSEM framework effectively models and exploits the scene-event relation for ASC, even using the information of events with unverified pseudo labels. 4) The weights of classification losses ($\mathrm{L_{scene}}$, $\mathrm{L_{event}}$) and regression losses ($\mathrm{L_{s\_by\_e}}$, $\mathrm{L_{e\_by\_s}}$) in cSEM can be further adjusted to achieve improved accuracy of ASC.

D. RQ4: How do different loss functions perform on losses related to the cooperative modelling of scene-event relation?

In the cSEM framework, the $\mathrm{loss_s}$ in $\mathrm{L_{scene}}$ defaults to the CE loss, the $\mathrm{loss_e}$ in $\mathrm{L_{event}}$ defaults to the BCE loss. The regression loss MSE [24] is adopted in cSEM to minimize the two-way relation-related losses, which is expected to fit the outputs of both branches further to align the knowledge of AS and AE. As described in Section II-C, the regression loss is utilized for the cooperative modelling of scene-event relations as it helps enhance the encoding of the whole output of embeddings, to improve the training of the coupling matrix. Based on such theoretical reasoning, MSE loss is used in preceding RQs. To confirm this reasoning, we also evaluate the performance achieved by other loss functions. Table IX shows the performance of using different loss functions for modelling the scene-event relation under the model structure with optimal weights of losses. Following the notations of Section II, $\hat{z}_s$ and $\hat{z}_e$ denote the logit [14] of $\hat{y}_s$ and $\hat{y}_e$, respectively.

TABLE IX
ACC. (%) OF ASC WITH DIFFERENT LOSS FUNCTIONS IN MODELLING SCENE-EVENT RELATION ON TUT2018 VALIDATION SET. $\mathcal{S}$ AND *Sig* DENOTE THE SOFTMAX AND SIGMOID FUNCTIONS, RESPECTIVELY.

#	$\mathrm{L_{s\_by\_e}}$	$\mathrm{L_{e\_by\_s}}$	cSEM-AST	cSEM-PANN
1	$CE(\mathcal{S}(\tilde{y}_s), Max(\hat{y}_s))$	$BCE(Sig(\tilde{y}_e), \hat{y}_e)$	76.35±1.95	72.33±1.83
2	$soft\ CE(\mathcal{S}(\tilde{y}_s), \hat{y}_s)$	$BCE(Sig(\tilde{y}_e), \hat{y}_e)$	77.34±2.09	73.38±1.39
3	$MSE(\tilde{y}_s, \hat{z}_s)$	$MSE(\tilde{y}_e, \hat{z}_e)$	82.06±2.32	76.97±1.41
4	$MSE(\tilde{y}_s, \hat{y}_s)$	$MSE(\tilde{y}_e, \hat{y}_e)$	82.15±2.48	77.48±1.72
5	$MSE(\tilde{y}_s, \log \hat{y}_s)$	$MSE(\tilde{y}_e, \hat{y}_e)$	**83.81**±1.89	**79.80**±1.64

In Table IX, #1 and #2 employ CE and BEC, respectively, to measure the distance between the predictions derived from the implicit scene-event relation and the actual outputs of the corresponding branch. Since AE in AEC tasks is typically considered independent of each other, BCE is applied for $\mathrm{L_{e\_by\_s}}$. The input and target in BCE generally default to a probability distribution, so the input of BCE uses Sigmoid as an activation function. Regarding CE related to the inferred scene, two types of CE are available: CE and soft CE [38], respectively. The targets of CE in classification usually consist of hard labels of 0 and 1, hence the probability distribution $\hat{y}_s$ is mapped to the one-hot vector by Max function to supervise the training process. For soft CE, the value of probability $\hat{y}_s$ is used as the soft target to preserve rich inter-class relation information. When using high-entropy soft targets, each training sample can be provided with more information than using hard targets, and the gradient between training samples has smaller variances [39]. Compared with CE, soft CE in #2 allows the non-target classes to be more prominent in training, leading to more reliable training and better results.

In Table IX, #3, #4 and #5 employ MSE to estimate inferred predictions. #3 directly calculates the distance between the logit vectors before activation functions. A previous study [40] about the gradient of CE shows that when the logit values are relatively small, the optimization effect of CE and MSE is equivalent. A comparison [24] of entropy-based loss and MSE reveals that MSE has better calibration abilities to correct the errors. It can be seen that #4 outperforms #3, probably because the activation functions such as Softmax and Sigmoid restrict the values of $\tilde{y}_s$ and $\tilde{y}_e$ to [0, 1], thus prune the target space to

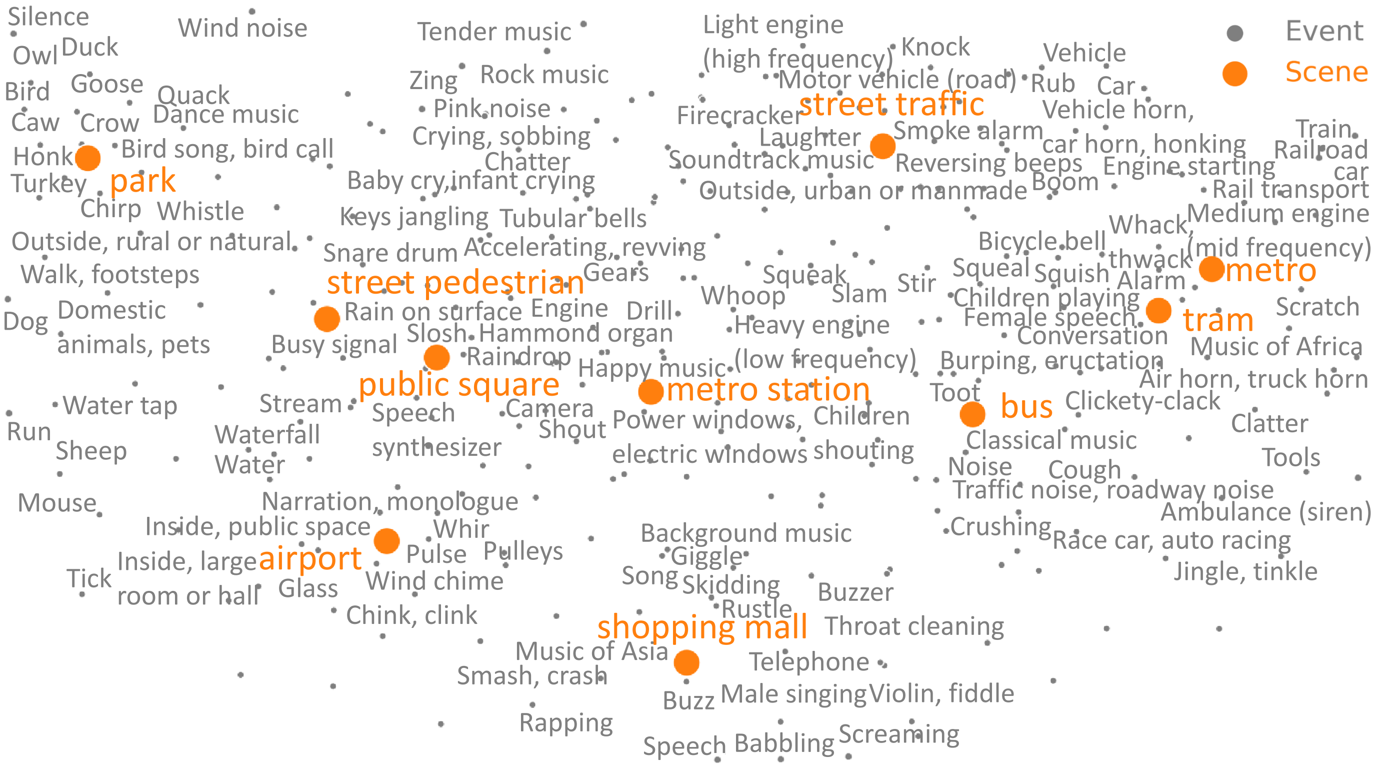


Fig. 4. Visualization of the core AS knowledge $\boldsymbol{W}_s$ and the core AE knowledge $\boldsymbol{W}_e$ learned by the proposed cSEM-AST using t-SNE [37].

a smaller range. This leads to the comparison of the original logits $\hat{z}_s$ and $\hat{z}_e$ as targets with $\tilde{y}_s$ and $\tilde{y}_e$ which are more likely to guide the model to find the optimal local solution in the target space. A drawback of Softmax in $\hat{y}_s$ is that by normalizing an exponential function, the largest value is highlighted, and the other values are suppressed significantly [41], resulting in larger gaps between similar values. Using log Softmax can alleviate this issue, and bring additional advantages [42], such as numerical stability and gradient additivity in training. After log mapping, the $\hat{y}_s$ as targets can result in an effect of model regularization, similar to label-smoothing regularization [43]. Log Softmax helps prevent the maximum value from becoming significantly larger than the remaining values. Log Softmax is used in #5 to make the target smoother and increase the entropy it implies, which achieves better results than #4. It may be because some scenes are similar, such as *buses* and *trams* during rush hours, *clamorous streets* and *noisy parks*. There are both similarities and subtle differences between these scenes. Using log Softmax, the difference between AS is less likely over-magnified, and thus reflects better the relation between similar scenes.

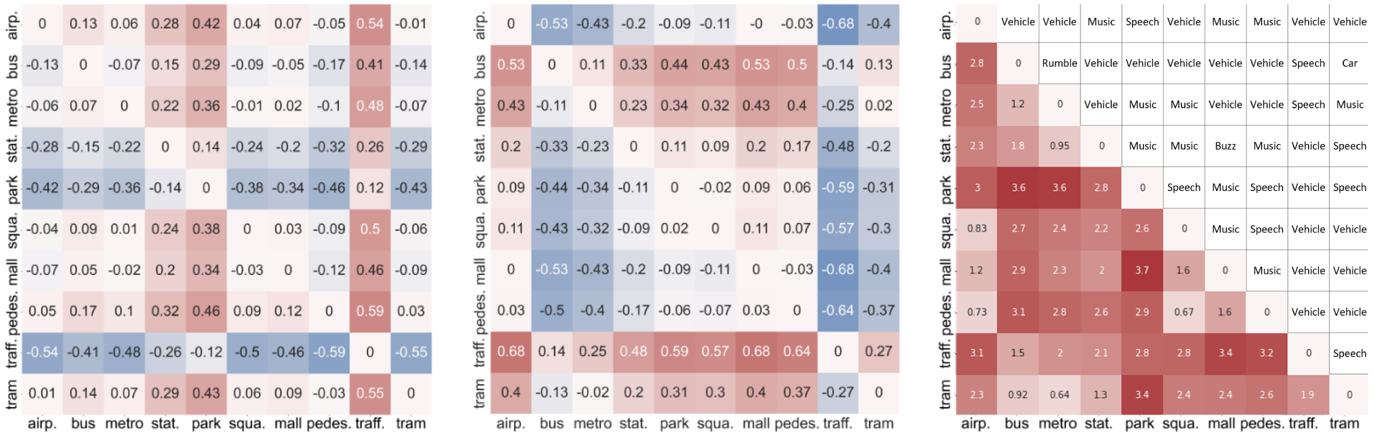
E. RQ5: Does the cooperative modelling of the implicit scene-event relations align the knowledge of AS and AE? Does it help reduce the overlap between their semantic spaces?

To provide intuitive insight into the implicit and intricate relation between AS knowledge $\boldsymbol{W}_s$ and AE knowledge $\boldsymbol{W}_e$, Fig. 4 shows the distribution of $\boldsymbol{W}_s$ belonging to the scene space and $\boldsymbol{W}_e$ belonging to the event space in the same latent space. In Fig. 4, AS and the unique AE contained in them are clustered together, for example, the *park* scene, and the events in this scene, such as *goose*, *crow*, and *bird song*. Events such as *music* and *buzz* often occur in the *shopping mall*. The *tram* is accompanied by events such as *scratch*, *conversation*, and *alarm*. On the contrary, AEs are not unique to one AS, such as *Baby cry* is more in between AS, indicating a lesser alignment between core knowledge in that area. The distribution in Fig. 4 reveals that (*public square*, *street pedestrian*) and (*metro*, *tram*) are closer AS pairs in the latent space, and these AS pairs are indeed similar in real life. Various events in Fig. 4 are clustered around the corresponding scenes orderly, demonstrating that the cSEM can align the semantic spaces of AS and AE by the two-way scene-event bridge via the coupling matrix.

TABLE X
AVERAGE RESULTS OF PCC AND MI BETWEEN $\boldsymbol{E}_s$ AND $\boldsymbol{E}_e$ ON TUT2018.

Dataset	Metric	MoE-AST	MIhE-AST	cSEM-AST
Training set	*PCC*	1.000	0.006	0.001
	MI (nat)	4.784	3.290	2.958
Testing set	*PCC*	1.000	0.002	0.001
	MI (nat)	4.827	3.423	2.983

Table X shows the correlation and overlap of AS and AE embeddings of audio samples in different frameworks, using Pearson Correlation Coefficient (PCC) [44] and e-based Mutual Information (MI) with the unit of nat [45]. Since the AS and AE embeddings used in MoE-AST are the same, its PCC equals 1, and MI is the largest in Table X. Compared to MIhE-AST, the PCC between AS and AE embeddings in cSEM-AST is reduced, and the MI is less. The reduction in PCC and MI between the learned AS and AE embeddings clarifies that the similarity between AS and AE representations learned by the



(a) Differences in *Speech* in different scenes. (b) Differences in *Vehicle* in different scenes. (c) Overall differences between scenes.

Fig. 5. Differences between acoustic scenes on the development set of TUT2018, where subplots (a) and (b) take the audio event *Speech* and *Vehicle* as examples, while subplot (c) shows the overall differences between scenes.

cSEM-AST is less, while the redundant information between AS and AE embeddings is reduced. The logical alignment of the core knowledge of AS and AE shown in Fig. 4 and the reduction of the redundant information of ASC and AEC branches in cSEM-AST allow the model to represent AS and AE for the classification tasks collaboratively.

F. RQ6: What are the differences among the ten classes of scenes? Which events contribute most to these differences?

This part analyses the distribution differences of AEs across scenes, and the overall differences between scenes when all AEs are considered together. AE classes are identified by 527 pseudo labels from AudioSet, that are used in training.

To analyse differences between AS in terms of individual AE, first, we obtain the probabilities of 527 classes of AE in all audio clips from the AEC branch of cSEM-AST. Then, the probabilities of 527 types of AE in audio clips classified in each scene class are summed and averaged. These averages are regarded as the overall probability of the corresponding event occurring in the scene. Finally, the mean of AE probabilities in different scenes is subtracted from each other to measure the difference of this event across scenes. Following this procedure, a probability difference matrix between scenes is obtained for each AE. Fig. 5 (a) and (b) illustrate these results for the AE *speech* and *vehicle*. In Fig. 5 (a), the probability of *speech* varies the most between two scenes: street pedestrian (*pedes.*) and street traffic (*traff.*). Fig. 5 (b) shows that the probability of vehicle differs the most between the shopping mall (*mall*) and *traff.* scenes. Thus, the presence of specific AE can clearly help to differentiate between scenes.

To explore whether all event classes taken together can help differentiate between scenes, the absolute values of the differences of all events are added to get the overall difference between the 10 classes of scenes. The result is shown in Fig. 5 (c). Since absolute values are used, the matrix in Fig. 5 (c) is symmetric. Hence, the AE contributing to the largest difference is shown in the upper-triangle part to indicate what events are mainly responsible for these differences. Fig. 5 (c) illustrates that among scenes, *mall* and *park* have the largest differences, with the main event causing such difference being *music*. The pair of scenes *park* and *bus*, and *park* and *metro* have the second-largest differences, and the main events causing these differences are *vehicles* and *music*, respectively. The difference between *bus* and *tram* is relatively small, and the audio event causing the difference is mainly *car*.

One can notice the resemblance between this probability difference matrix in Fig. 5 (c) and the confusion matrix in Fig. 3 (f). As expected, the proposed cSEM-AST shows lower confusion between AS classes where AE probabilities differ most. Based on Fig. 5 (c), it is easy to identify the main events contributing to these differences. That means, the proposed cSEM-AST works well for joint scene-event analysis.

G. RQ7: How do cSEM-based methods compare to others?

Table XI shows the ASC results of different methods. The CNN-based ensemble of two multi-input CNN models trained with 11 types of acoustic features has achieved 1st place in the challenge of DCASE2018 task 1 A (T1A) [46]. Also in T1A, the system in 2nd place [47] uses a depth-wise separable CNN trained with 3 multiscale features. The system in 3rd place [48] is an ensemble of 6 big and deep models trained with 4 types of acoustic features. In contrast, the proposed cSEM-AST and cSEM-PANN only use one type of feature with one model and do not involve data augmentations. A simple PANN-based hierarchical baseline with an upper-lower relationship between AE and AS prediction layers is proposed to estimate AE and AS with an explicitly formed hierarchy. To explore the performance of the linear combination of activation maps-based cross-stitch [9], we show the CNN-based cross-stitch-PANN in Table XI. Furthermore, several AE embeddings are used as nodes to build the AS graph representations in the event relational graph representation learning [49]. Note that external event and scene datasets are allowed in T1A, so most of the above methods have used models trained or pretrained on these external datasets. Compared with the above methods, the proposed cSEM, which aims to exploit the implicit two-way scene-event relation to improve the discriminative power between similar AS, achieves a better result. This demonstrates that scene-event relation modelling helps improve scene classification, even if the event information is derived from unverified noisy pseudo labels.

TABLE XI
Comparison of the ASC results on TUT2018 test set.

System	Model Cornerstone	*Acc. (%)*
Baseline [15]	CNN	59.7
Simple hierarchical baseline	VGG-like CNN	72.9
Cross-stitch-PANN [9]	VGG-like CNN	74.5
Dataset-specific relation RGASC [8]	VGG-like CNN	77.4
Event relational graph [49]	CNN & Gated GCN	78.1
Wavelet-based DSS [20]	CRNN with attention	78.3
T1A-3rd (6 models×4 features) [48]	VGG & ResNet	78.4
Proposed cSEM-PANN	VGG-like CNN	78.5
T1A-2nd (3 multiscale features) [47]	Separable CNN	79.8
T1A-1st (2 models×11 features) [46]	Multi-inputs CNN	80.1
Proposed cSEM-AST	Transformer Encoder	**81.0**

TAU2019 from DCASE2019 T1A [50] is an expanded acoustic scene dataset based on TUT2018 [30]. Given the good performance of cSEM-AST in previous RQs, Table XII presents the results of cSEM-AST on TAU2019. DCASE2019 T1A allows the use of external datasets to train models. Therefore, in Table XII, cSEM-AST is pretrained on AudioSet, and the public and private scene datasets of DCASE2013 [51]. The parameters used in pretraining are consistent with those used in Section IV. SpecAugment [52] and Mixup [53] are used for data augmentation. Since some labels of DCASE2013 and TAU2019 datasets are different, we manually map *busy street* and *quiet street* in DCASE2013 to *street traffic* in TAU2019, *open air market* and *supermarket* to *shopping mall*, *tube* to *metro*, and *tube station* to *metro station*. Labels that are the same in these two datasets are retained. Given the advantages of the multi-channel methods [54][55] in DCASE2019 T1A, cSEM-AST also uses log mel features on the left channel, the right channel, and the difference between the two channels. Finally, the cSEM-AST achieves competitive results compared to other multi-model fusion or ensemble methods in Table XII.

TABLE XII
Comparison of the ASC results on TAU2019 test set.

System	Model Cornerstone	*Acc. (%)*
Baseline [15]	CNN	62.5
Attentive fusion CNN (3 models) [56]	VGG	77.0
Clustered DNN (9 models×3 features) [57]	VGG-like CNN	81.6
DCGAN (3 models, 3 features) [58]	VGG	88.0
Multi-resolution DSS (2 channels) [55]	CNN-GRU	88.1
CNN vote (4 models×3 channels) [54]	ResNet	88.4
Proposed cSEM-AST (3 channels)	Transformer Encoder	**88.9**

In addition, Table XIII compares cSEM with other scene-event joint analysis methods. Since AS and AE in the synthesized dataset are not as complex as those in real life, the models in Table XIII usually offer better results on JSSED. Among them, the joint scene and event recognition [5] by the same embedding space gives the lowest accuracy. This is probably because real-life coarse-grained scenes and fine-grained events have their own characteristics and attributes. The performance of jointly analyses AS and AE based on one-way scene-to-event conditional loss [7] is better than that of [6], due to the use of the scene-conditioned loss. Overall, the proposed cSEM-based model provides the best scores out of the discussed methods for joint analysis of AS and AE.

TABLE XIII
Comparison of ASC results for scene-event analysis methods.

Dateset	Scene-event joint analysis system	*Acc. (%)*
TUT2018	Joint scene and event recognition [5]	52.4
	Event and scene joint analysis using MTL [6]	61.7
	Conditional scene and event recognition [7]	66.4
	Proposed cSEM-AST	**81.0**
JSSED	Joint scene and event recognition [5]	92.0
	Event and scene joint analysis using MTL [6]	93.7
	Conditional scene and event recognition [7]	93.9
	Proposed cSEM-AST	**97.2**

VI. Conclusion

This paper has presented a new method for modelling the intrinsic relations between audio scenes and events using automatically learned coupling matrices, and using such relations to improve ASC. The proposed cSEM framework facilitates the alignment of the information from coarse-grained AS and fine-grained AE, and helps to reduce the confusion between similar AS, thus further improving ASC performance. Experiments show that: 1) sharing some layers in cSEM-based models will improve their performance; 2) The cSEM improves the accuracy of Transformer-based, CNN-based, and CNN-Transformer-based models on ASC. Compared with MoE and MlhE frameworks, the cSEM framework further reduces the confusion between similar scenes; 3) The cSEM improves ASC performance by associating the information of AS and AE, even if the information of AE is derived from unverified pseudo-labels. Specifically, cSEM improves the accuracy of cSEM-AST and cSEM-PANN on ASC by 3.97% and 4.15%, respectively. 4) In cSEM, the regression loss is more effective than the classification loss for the cooperative modelling of scene-event relations; 5) The cSEM can help align the knowledge of AS and AE through the coupling matrix, and reduce redundant information between AS and AE embeddings; 6) The cSEM-based model works well in capturing the differences between scenes from the perspective of events in real-life scene-event analysis; 7) Compared with other multi-feature or multi-model ensemble methods, the cSEM-based model achieves competitive results on ASC. The Acc. of ASC on TUT2018, TAU2019 and JSSED datasets are 81.0%, 88.9% and 97.2%, respectively. The proposed cSEM contains four loss functions, and future work will explore how to automatically adjust the weights of loss functions to adapt to cSEM with different structures.

References

[1] S. Souli and Z. Lachiri, "Audio sounds classification using scattering features and support vectors machines for medical surveillance," *Applied Acoustics*, vol. 130, pp. 270–282, 2018.

[2] Y. Hou, Z. Yu, X. Liang, X. Du, B. Zhu, Z. Ma, et al., "Attention-based cross-modal fusion for audio-visual voice activity detection in musical video streams," in *Proc. of INTERSPEECH*, 2021, pp. 321–325.

[3] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, et al., "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 379–393, 2018.

[4] D. Botteldooren, T. Andringa, I. Aspuru, A. L. Brown, D. Dubois, C. Guastavino, J. Kang, et al., "From sonic environment to soundscape," *Soundscape and the Built Environment*, vol. 36, pp. 17–42, 2015.

[5] H. L. Bear, I. Nolasco, and E. Benetos, "Towards joint sound scene and polyphonic sound event recognition," in *Proc. of INTERSPEECH*, 2019, pp. 1236–1240.

[6] N. Tonami, K. Imoto, R. Yamanishi, et al., "Joint analysis of sound events and acoustic scenes using multitask learning," *IEICE Transactions on Information and Systems*, vol. 104, no. 2, pp. 294–301, 2021.

[7] T. Komatsu, K. Imoto, and M. Togami, "Scene-dependent acoustic event detection with scene conditioning and fake-scene-conditioned loss," in *Proc. of ICASSP*, 2020, pp. 646–650.

[8] Y. Hou, B. Kang, W. Van Hauwermeiren, and D. Botteldooren, "Relation-guided acoustic scene classification aided with event embeddings," in *Proc. of IJCNN*, 2022, pp. 1–8.

[9] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proc. of CVPR*, 2016, pp. 3994–4003.

[10] K. Imoto, N. Tonami, Y. Koizumi, M. Yasuda, et al., "Sound event detection by multitask learning of sound events and scenes with soft scene labels," in *Proc. of ICASSP*, 2020, pp. 621–625.

[11] Y. Cao, T. Iqbal, Q. Kong, F. An, W. Wang, and M. D. Plumbley, "An improved event-independent network for polyphonic sound event localization and detection," in *Proc. of ICASSP*, 2021, pp. 885–889.

[12] N. Tonami, K. Imoto, M. Niitsuma, R. Yamanishi, and Y. Yamashita, "Joint analysis of acoustic events and scenes based on multitask learning," in *Proc. of WASPAA*, 2019, pp. 338–342.

[13] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *Proc. of ICASSP*, 2018, pp. 121–125.

[14] L. J. Ba and R. Caruana, "Do deep nets really need to be deep?," in *Proc. of NIPS*, 2014, pp. 2654–2662.

[15] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proc. of DCASE2018*, 2018, pp. 9–13.

[16] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[17] A. Kumar and B. Raj, "Audio event detection using weakly labeled data," in *Proc. of ACM MM*, 2016, pp. 1038–1047.

[18] Y. Hou, Q. Kong, J. Wang, and S. Li, "Polyphonic audio tagging with sequentially labelled data using crnn with learnable gated linear units," in *Proc. of DCASE2018*, 2018, pp. 78–82.

[19] Y. Hou, Q. Kong, S. Li, and M. D. Plumbley, "Sound event detection with sequentially labelled data based on connectionist temporal classification and unsupervised clustering," in *Proc. of ICASSP*, 2019, pp. 46–50.

[20] Z. Li, Y. Hou, X. Xie, S. Li, L. Zhang, S. Du, and W. Liu, "Multi-level attention model with deep scattering spectrum for acoustic scene classification," in *Proc. of ICMEW*, 2019, pp. 396–401.

[21] H. Jallet, E. Cakır, and T. Virtanen, "Acoustic scene classification using convolutional recurrent neural networks," Tech. Rep., DCASE2017 Challenge, 2017.

[22] Y. Hou, F. K. Soong, J. Luan, and S. Li, "Transfer learning for improving singing-voice detection in polyphonic instrumental music," in *Proc. of INTERSPEECH*, 2020, pp. 1236–1240.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, et al., "Attention is all you need," in *Proc. of NIPS*, 2017, pp. 5998–6008.

[24] T. Kim, J. Oh, N. Y. Kim, S. Cho, and S. Yun, "Comparing Kullback-Leibler divergence and mean squared error loss in knowledge distillation," in *Proc. of IJCAI*, 2021, pp. 2628–2635.

[25] C. Wang, J. Xiao, Y. Han, et al., "Towards learning spatially discriminative feature representations," in *Proc. of ICCV*, 2021, pp. 1326–1335.

[26] M. Lim, D. Lee, H. Park, Y. Kang, J. Oh, J. Park, et al., "Convolutional neural network based audio event classification," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 6, pp. 2748–2760, 2018.

[27] Y. Gong, Y. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," in *Proc. of INTERSPEECH*, 2021, pp. 571–575.

[28] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, et al., "AudioSet: An ontology and human-labeled dataset for audio events," in *Proc. of ICASSP*, 2017, pp. 776–780.

[29] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, "A comparison of transformer and lstm encoder decoder models for ASR," in *Proc. of ASRU*, 2019, pp. 8–15.

[30] T. Heittola, A. Mesaros, and T. Virtanen, "TAU urban acoustic scenes 2019, development dataset," Mar. 2019.

[31] Y. Hou, Y. Wang, W. Wang, et al., "GCT: Gated contextual transformer for sequential audio tagging," in *Proc. of ICASSP*, 2023, pp. 1–5.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of ICLR*, 2015.

[33] Q. Kong, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, "DCASE 2018 challenge surrey cross-task convolutional neural network baseline," in *Proc. of DCASE2018*, 2018, pp. 217–221.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, et al., "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[35] B. Zhang, S. Zhang, and W. Li, "Bearing performance degradation assessment using long short-term memory recurrent network," *Computers in Industry*, vol. 106, pp. 14–29, 2019.

[36] C. Cortes and M. Mohri, "AUC optimization vs. error rate minimization," in *Proc. of NIPS*, 2003, vol. 16.

[37] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[38] L. Feng, S. Shu, Z. Lin, F. Lv, L. Li, and B. An, "Can cross entropy loss be robust to label noise?," in *Proc. of IJCAI*, 2020, pp. 2206–2212.

[39] T. Chen, W. Wu, Y. Gao, L. Dong, X. Luo, and L. Lin, "Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding," in *Proc. of ACM MM*, 2018, pp. 2023–2031.

[40] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, 2015.

[41] Y. Luo, Y. Wong, et al., "G-softmax: improving intraclass compactness and interclass separability of features," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 2, pp. 685–699, 2019.

[42] S. Kanai, Y. Fujiwara, Y. Yamanaka, et al., "Sigsoftmax: reanalysis of the softmax bottleneck," in *Proc. of NIPS*, 2018, pp. 284–294.

[43] C. Szegedy, V. Vanhoucke, et al., "Rethinking the inception architecture for computer vision," in *Proc. of CVPR*, 2016, pp. 2818–2826.

[44] J. Adler and I. Parmryd, "Quantifying colocalization by correlation: the pearson correlation coefficient is superior to the mander's overlap coefficient," *Cytometry Part A*, vol. 77, no. 8, pp. 733–742, 2010.

[45] T. O. Kvålseth, "On normalized mutual information: measure derivations and properties," *Entropy*, vol. 19, no. 11, pp. 631, 2017.

[46] A. Golubkov and A. Lavrentyev, "Acoustic scene classification using convolutional neural networks and different channels representations and its fusion," Tech. Rep., DCASE2018 Challenge, 2018.

[47] L. Yang, X. Chen, and L. Tao, "Acoustic scene classification using multi-scale features," Tech. Rep., DCASE2018 Challenge, 2018.

[48] O. Mariotti, M. Cord, et al., "Exploring deep vision models for acoustic scene classification," Tech. Rep., DCASE2018 Challenge, 2018.

[49] Y. Hou, S. Song, C. Yu, Y. Song, W. Wang, et al., "Multi-dimensional edge-based audio event relational graph representation learning for acoustic scene classification," *arXiv preprint arXiv:2210.15366*, 2022.

[50] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification in DCASE 2019 challenge: Closed and open set classification and data mismatch setups," in *Proc. of DCASE2019*, 2019, pp. 164–168.

[51] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[52] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, et al., "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. of INTERSPEECH*, 2019, pp. 2613–2617.

[53] S. Wei, K. Xu, D. Wang, F. Liao, H. Wang, and Q. Kong, "Sample mixed-based data augmentation for domestic audio tagging," in *Proc. of DCASE2018*, 2018, pp. 93–97.

[54] M. Liu and Y. Li, "The system for acoustic scene classification using resnet," Tech. Rep., DCASE2019 Challenge, 2019.

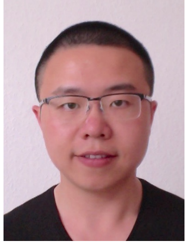
[55] S. Ma and W. Liu, "Acoustic scene classification based on binaural deep scattering spectra with neural network," Tech. Rep., DCASE2019 Challenge, 2019.

[56] H. Zeinali, L. Burget, and H. Cernocky, "Acoustic scene classification using fusion of attentive convolutional neural networks for dcase2019 challenge," Tech. Rep., DCASE2019 Challenge, 2019.

[57] M. Plata, "Deep neural networks with supported clusters preclassification procedure for acoustic scene recognition," Tech. Rep., DCASE2019 Challenge, 2019.

[58] F. Ning, S. Duan, P. Han, J. Wei, and Z. Ding, "Acoustic scene classification based on the dataset with deep convolutional generated against network," Tech. Rep., DCASE2019 Challenge, 2019.

Yuanbo Hou is pursuing the Ph.D. degree in Computer Science Engineering at the WAVES Research Group, Ghent University, Belgium, under the supervision of Prof. Dick Botteldooren. He received the M.E. degree from the Beijing University of Posts and Telecommunications, China, in 2020. He worked as a short-term Honorary Research Assistant at University College London, U.K., in 2022, under the supervision of Prof. Jian Kang. His research concerns AI for sound, using deep learning and signal processing to analyze and recognize sounds. He serves as a reviewer for IEEE/ACM Transactions on Audio Speech and Language Processing, and International Conference on Acoustics, Speech, and Signal Processing (ICASSP; 2022-).

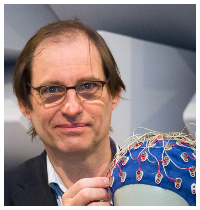


Bo Kang is a postdoctoral researcher at the IDLab, Ghent University, Belgium. He holds a Ph.D. degree in Computer Science Engineering from Ghent University, Belgium. His primary interests are data mining and machine learning, and more specifically representation learning and dimensionality reduction. He has a website at http://bokang.io.

Andrew Mitchell is a research Fellow in urban soundscape modelling at University College London, U.k. He earned his BSc (Hons) in Physics Music at Cardiff University from 2012-2015. He received the Ph.D. degree from University College London in 2022. He worked as an acoustical consultant dealing with noise impacts from wind farms in Wales and England before moving to Los Angeles to focus on environmental and architectural acoustics. Throughout his Ph.D. he has continued working with engineering consultancies, including London-based Hoare Lea, to bring a more holistic soundscape approach in acoustics engineering and design. Andrew's research is highly collaborative, resulting in him working and publishing with teams at Stockholm University, University of Granada, La Salle URL (Barcelona), and Université Gustave Eiffel.

Wenwu Wang is a Professor in Signal Processing and Machine Learning, and a Co-Director of the Machine Audition Lab within the Centre for Vision Speech and Signal Processing, University of Surrey, UK. He is also an AI Fellow at the Surrey Institute for People Centred Artificial Intelligence. His current research interests include signal processing, machine learning and perception, artificial intelligence, machine audition (listening), and statistical anomaly detection. He has (co)-authored over 350 papers in these areas. He has been involved as Principal or Co-Investigator in more than 30 research projects, funded by UK and EU research councils, and industry (e.g. BBC, NPL, Samsung, Tencent, Huawei, Saab, Atlas, and Kaon). He is the elected Chair of IEEE Signal Processing Society (SPS) Machine Learning for Signal Processing Technical Committee, the Vice Chair of the EURASIP Technical Area Committee on Acoustic Speech and Music Signal Processing, a Board Member of IEEE SPS Technical Directions Board. He is an Associate Editor for IEEE/ACM Transactions on Audio Speech and Language Processing, an Associate Editor for (Nature) Scientific Report, and a Specialty Editor in Chief of Frontier in Signal Processing. He was a Senior Area Editor (2019-2023) and an Associate Editor (2014-2018) for IEEE Transactions on Signal Processing. He has been an invited keynote or plenary speaker on more than 20 international conferences and workshops, and a member of the technical program committee for more than 100 international conferences or workshops.

Jian Kang is a Professor of acoustics and soundscape at the University College London. He has worked in the field for 40 years, with 800+ publications. He is President of the International Institute of Acoustics and Vibration (IIAV), and he also chairs the European Acoustics Association Technical Committee for Noise, and the EU COST Action on Soundscape of European Cities and Landscapes. He is Fellow of Royal Academy of Engineering, and a Member of Academia Europaea -The Academy of Europe.



Dick Botteldooren is a full Professor at Ghent University, where he leads research on Acoustics and teaches a variety of courses related to sound and computational methods. He obtained a MSc degree in Electronic Engineering in 1986 from Ghent University and a PhD in Applied Science in 1990 from Ghent University. In 1992, he became interested in acoustics and, in particular, environmental sound. Dick Botteldooren is currently the president of the European Acoustics Association. Between 2004 and 2013, he was the Editor-in-Chief of Acta Acustica united with Acustica, the journal of the European Acoustics Association. Until 2018, he was the president of the Belgian Acoustical Society; between 2015 and 2018, he was I-INCE vice-president for Europe and Africa. He is a fellow of the Acoustical Society of America and the Institute of Acoustics and Vibration. Dick Botteldooren has made research contributions in the field of acoustic modeling, noise mapping, environmental sensor networks, computational intelligence, modeling perception of environmental sound, health impacts of sound, biomonitoring, urban sound planning, soundscapes, and noise policy support. His work was reported in approximately 200 journal publications and several hundred conference contributions. Based on his expertise, he was an advisor for national and international health councils and noise policymakers.